

VetCoders / mcp-server-semgrep Public[Code](#) [Issues](#) [Pull requests](#) [Discussions](#) [Actions](#) [Projects](#) [Security](#)

# fix(security): patch CWE-78 OS command injection + trampoline cleanup #15

MergedSzowesgad merged 10 commits into `main` from `fix/command-injection-and-path-va...`

last week

[Conversation](#) 6[Commits](#) 10[Checks](#) 0[Files changed](#) 18Szowesgad commented [2 weeks ago](#)Collaborator

## Summary

Closes [#12](#) — OS command injection (CWE-78, CVSS 9.8) reported by [@BruceJqs](#).

Supersedes [#13](#) and [#14](#). The CWE-78 fix from [#14](#) is included via merge commit ( `78030da` , `26098f3` ); the defense-in-depth concept proposed by [@xyaz1313](#) in the [#12](#) thread is extended into `validateNoShellMetacharacters` .

## What changed

### Security

- All `child_process.exec()` calls replaced with `execFile()` (no shell). Affects `handleScanDirectory` , `checkSemgrepInstallation` , `installSemgrep` .
- All shell-based file IO ( `cat / echo > file` ) replaced with `fs.promises.readFile / writeFile` in `analyze_results` , `filter_results` , `export_results` , `compare_results` , `create_rule` .
- `validateNoShellMetacharacters` wired into `validateAbsolutePath` as defense-in-depth — rejects `;&\ $()<>{}!#*?[]"~`` and whitespace control characters before any value reaches the filesystem.
- `handleCreateRule` hardened: allowlists for `id` , `language` , `severity` (fail-closed) + `escapeYamlScalar` ( `JSON.stringify` ) for `pattern / message` — closes YAML injection vector flagged by Gemini Code Assist on [fix: prevent OS command injection \(CWE-78\) in shell exec calls #14](#).
- `SEMGREP_APP_TOKEN` redacted via position-based mapping in stderr logs.

- 50 MiB `maxBuffer` cap on `execFile` `semgrep` calls (prevents truncation on large JSON output — also flagged by Gemini Code Assist).

## Trampoline cleanup

- Bump 1.0.0 → 1.0.1.
- Repository / homepage / smithery URLs migrated to VetCoders org.
- Drop unused `axios` runtime dependency.
- Delete dead `src/config.ts` (was never imported by the monolith).
- Replace stale `tests/handlers.test.ts` and `tests/utills.test.ts` (importing modules that never existed) with `tests/security.test.ts`: 43 cases including CWE-78 regression payloads (`'`; `id >&2`; `#``, backticks, pipes, YAML injection in language field).
- Add `SECURITY.md` with disclosure policy and operator hardening notes.
- Add `CHANGELOG.md`.

## Contribution credit

Contributor	Role
<a href="#">@BruceJqs</a>	Vulnerability discovery + CodeQL report ( <a href="#">#12</a> )
<a href="#">@karthikeyansundaram2</a>	CWE-78 fix foundation merged via <a href="#">#14</a> ( <code>'78030da`</code> , <code>'26098f3`</code> )
<a href="#">@xyaz1313</a>	<code>'validateNoShellMetacharacters`</code> concept proposal in <a href="#">#12</a> thread, extended in this PR
Gemini Code Assist	Flagged token leak regression and YAML injection follow-up under <a href="#">#14</a> review
<a href="#">@m-szymanska</a>	Repo co-maintainer

## Test plan

- `'tsc`` clean
- `'vitest run`` → 43/43 pass (validators + CWE-78 regression payloads)
- Smoke test: `'node build/index.js`` boots stdio
- Live MCP test: 4 attack payloads via `tools/call`` (`'analyze_results``, `'filter_results``, `'create_rule``, relative path) → all blocked by `'validateAbsolutePath`` before reaching shell
- Maintainer end-to-end test post-merge

## Next

- Publish GHSA advisory once merged
- Bump npm to 1.0.1

- Re-register Glama listing under VetCoders org (👍 [Glama listing is missing Dockerfile #10](#))
- Close 🚫 [fix: OS command injection vulnerability \(CWE-78\) in all handlers #13](#) and 🚫 [fix: prevent OS command injection \(CWE-78\) in shell exec calls #14](#) with thank-you comments referencing this PR

VibeCrafted with AI Agents (c)2026 VetCoders

👁️ 1

📁 **karthikeyansundaram2** and others added 4 commits [2 weeks ago](#)

- 🔗 [fix: prevent OS command injection \(CWE-78\) in shell exec calls](#) ... [26098f3](#)
- 🔗 [fix: address code review feedback on CWE-78 PR](#) ... [78030da](#)
- 🔗 [Closes #12 \(reported by BruceJin / @BruceJqs\).](#) ... [8b4ea5e](#)
- 🔗 [Prevent shell and YAML injection in semgrep server](#) ... [fcc696c](#)

👁️ **Szowesgad** requested a review from **m-szymanska** [2 weeks ago](#)

📁 **Szowesgad** mentioned this pull request [2 weeks ago](#)

### [Glama listing is missing Dockerfile #10](#)

🔒 Closed

**Szowesgad** commented [2 weeks ago](#)

Collaborator

Author

[@xyaz1313](#) [@karthikeyansundaram2](#) — your work is the foundation of this PR.

GitHub blocks formal review requests from non-collaborator forks, so I can't put you in the official review queue. But your fingerprints are on this commit:

- [@karthikeyansundaram2](#) — your CWE-78 fix from 🚫 [fix: prevent OS command injection \(CWE-78\) in shell exec calls #14](#) ( [78030da](#) , [26098f3](#) ) is included via the merge commit [fcc696c](#) . Co-Authored-By tag in place. The two educational comments you added ( `// Use execFile with arg array...` and `// Use JSON.stringify to escape...` ) survived line-by-line conflict resolution.
- [@xyaz1313](#) — your `validateNoShellMetacharacters` sketch from the [#12 thread](#) made it into the tree, extended with `*?["'~\t` and a few edge cases. Co-Authored-By tag in place.

If either of you has a moment to glance at the diff and leave a `:thumbsup:` or comments, that would be hugely appreciated. No obligation — your code is already credited regardless.

Thanks again — both PRs were the catalyst here.

— [@Szowesgad](#)

 **gemini-code-assist** (Bot) reviewed [2 weeks ago](#)

[View reviewed changes](#)

 **gemini-code-assist** (Bot) left a comment

## Code Review

This pull request updates the server to version 1.0.1 to address security vulnerabilities, specifically CWE-78 (OS Command Injection). It replaces shell-dependent executions with `execFile` and `fs/promises`, introduces comprehensive path and rule validation, and adds security-focused tests. Feedback indicates that the shell metacharacter regex may break Windows compatibility, the path validation logic needs to be more robust against prefix-based bypasses, and JSON parsing requires safer handling to avoid crashes on null inputs.

▼ `src/index.ts` Outdated

```
21 +
22 + const SEMGREP_MAX_BUFFER = 50 * 1024 * 1024;
23 +
24 + const SHELL_METACHARACTERS = /[;&|`$(<>{}\\!#\n\r\t*?[\]"'~]/;
```

 **gemini-code-assist** (Bot) [2 weeks ago](#)

**High Priority**

The `SHELL_METACHARACTERS` regex is overly aggressive and will cause the server to fail on Windows. Specifically, the inclusion of `\\` will cause `validateNoShellMetacharacters` to reject all absolute Windows paths (e.g., `C:\path\to\file`), as `path.normalize` uses backslashes as separators. Additionally, characters like `#`, `!`, and `~` are valid in many filesystem paths and do not pose a security risk when using `execFile` or `fs` APIs, which bypass the shell. Consider relaxing this regex to only include characters that are truly dangerous in the specific contexts where they are used.

Suggested change

```
- const SHELL_METACHARACTERS = /[;&|`$(<>{}\\!#\n\r\t*?[\]"'~]/;
```

```
+ const SHELL_METACHARACTERS = /[;&|`$()<>{}\\n\\r\\t"']/;
```

src/index.ts Outdated

Comment on lines +65 to +70

```
65 + if (!normalizedPath.startsWith(BASE_ALLOWED_PATH)) {
66 +   throw new McpError(
67 +     ErrorCode.InvalidParams,
68 +     `${paramName} must be within the MCP directory (${BASE_ALLOWED_PATH})`
69 +   );
70 + }
```



**gemini-code-assist** Bot 2 weeks ago

**Security** **High Priority**

The `startsWith` check is insufficient for path validation as it can be bypassed by directories that share a common prefix. For example, if `BASE_ALLOWED_PATH` is `/home/user/app`, a path like `/home/user/app-secret/config.json` would pass this check. Using `path.relative` is a more robust way to ensure the resolved path is strictly contained within the allowed base directory.

Suggested change

```
156 - if (!normalizedPath.startsWith(BASE_ALLOWED_PATH)) {
157 -   throw new McpError(
158 -     ErrorCode.InvalidParams,
159 -     `${paramName} must be within the MCP directory
160 -     (${BASE_ALLOWED_PATH})`
161 -   );
162 - }

156 + const relative = path.relative(BASE_ALLOWED_PATH, normalizedPath);
157 + if (relative.startsWith('.') || path.isAbsolute(relative)) {
158 +   throw new McpError(
159 +     ErrorCode.InvalidParams,
160 +     `${paramName} must be within the MCP directory
161 +     (${BASE_ALLOWED_PATH})`
162 +   );
163 + }
```

src/index.ts Outdated

422 | -

```

423 | - // Simple analysis of the results
    | +
399 | + const fileContent = await readFile(resultsFile, 'utf-8');
400 | + const results = JSON.parse(fileContent);

```

**gemini-code-assist** (Bot) 2 weeks ago

**Medium Priority**

`JSON.parse` can return `null` if the input string is `"null"`. If this happens, subsequent property access (like `results.results` on line 403 or 408) will throw a `TypeError` and crash the tool execution. This pattern occurs in several handlers (`handleAnalyzeResults`, `handleFilterResults`, `handleExportResults`, `handleCompareResults`). Consider adding a fallback or a type check.

Suggested change

```

- const results = JSON.parse(fileContent);
+ const results = JSON.parse(fileContent) || {};

```

**Szowesgad** and others added 2 commits 2 weeks ago

chore(npm): add files array + prepublishOnly gate [d3c9fb7](#)

chore(deps): move SDK to runtime, upgrade typescript-eslint to ^8.58.2 [d8ed43e](#)

**m-szymanska** previously approved these changes [last week](#)

[View reviewed changes](#)

**m-szymanska** left a comment Collaborator

Looks good to me overall!  
This is a real fix, not just patch-for-the-sake-of-patching.

Only thing I'd still tighten a bit is the `startsWith(BASE_ALLOWED_PATH)` check, because prefix-based checks can be a little too loose. `path.relative` would probably make that safer.

Other than that, direction looks very good!

fix(review): resolve PR #15 Gemini findings [ca50d8a](#)

✕ Szowesgad dismissed m-szymanska's stale review via ca50d8a last week

📁 Szowesgad added 3 commits last week

🔗 test(review): add stdio workspace-root smoke e716817

🔗 Optional SEMGREP token & server lifecycle handling ... ✕ bc33fe2

🔗 Install Semgrep via pip and add OSS fallback ... 761b519

🔗 Szowesgad merged commit 000603f into main last week

📁 This was referenced last week

**fix: OS command injection vulnerability (CWE-78) in all handlers #13**

🔒 Closed

**fix: prevent OS command injection (CWE-78) in shell exec calls #14**

🔒 Closed

🔗 Szowesgad deleted the fix/command-injection-and-path-validation branch 3 days ago

Sign up for free to join this conversation on GitHub. Already have an account? [Sign in to comment](#)

Reviewers

👤 m-szymanska

+1 more reviewer

🌟 gemini-code-assist[bot]

Assignees

No one assigned

Labels

None yet

---

### Projects

None yet

---

### Milestone

No milestone

---

### Development

Successfully merging this pull request may close these issues.

 **[bug] Security: OS Command Injection Vulnerability in mcp-server-semgrep**

---

### 3 participants

